You, Me & SVG!

Level 2

Would You, Could You With a Badge?

Section 1 – Circles by the Ton

Drawing a New Icon

Let’s create the icon for our Schmuffle screen!

=

We will start by including the file...

index.html

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <h1>Schmuffle-Screen Icon</h1>
    <img src="phone_icon.svg"/>
  </body>
</html>
```

phone_icon.svg

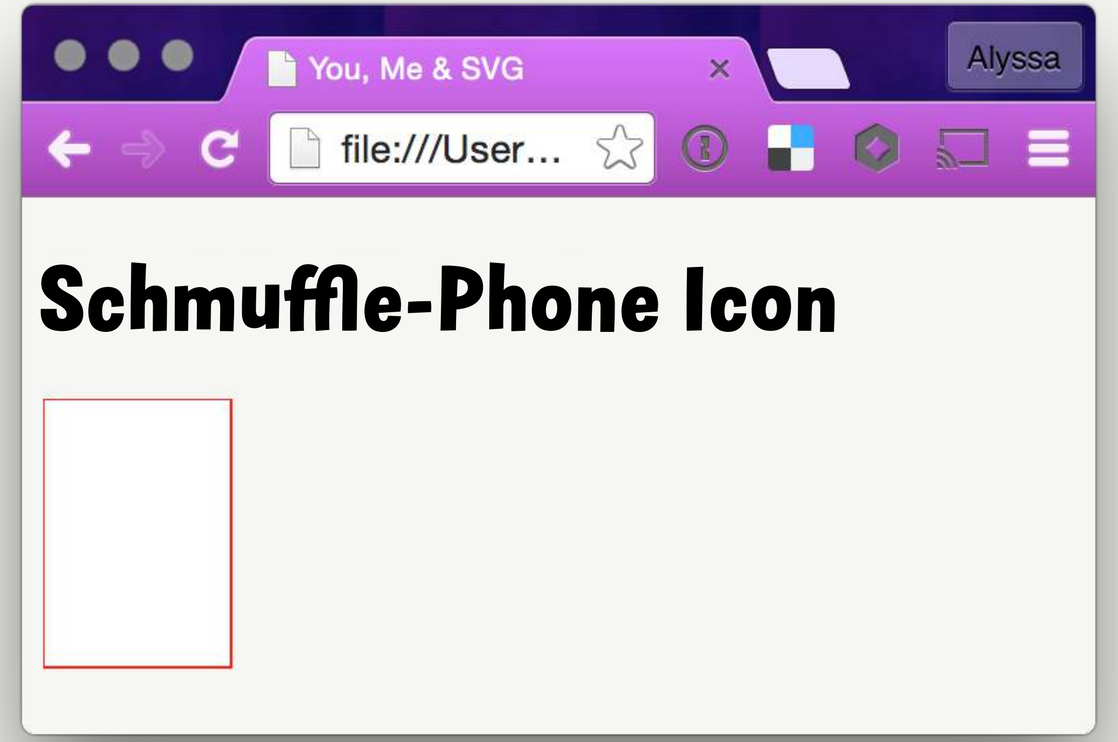
```
<svg...>
 <rect/>
</svg>
```

... and creating a rectangle.

Adding a Rectangle Stroke

We add a stroke color and we get a 1px outline.

```
<rect height="100" width="70" fill="white" stroke="#FF2626"/>
```

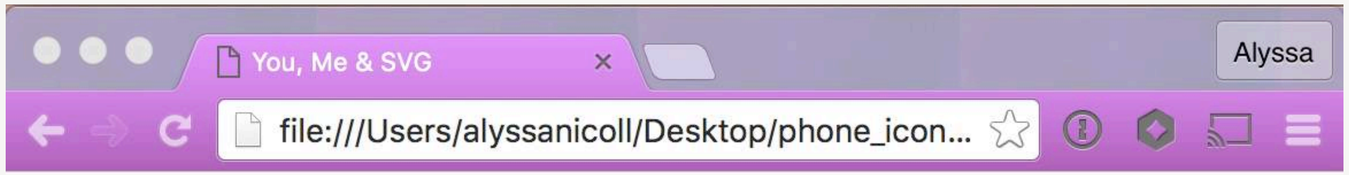


Creating a Thicker Outline

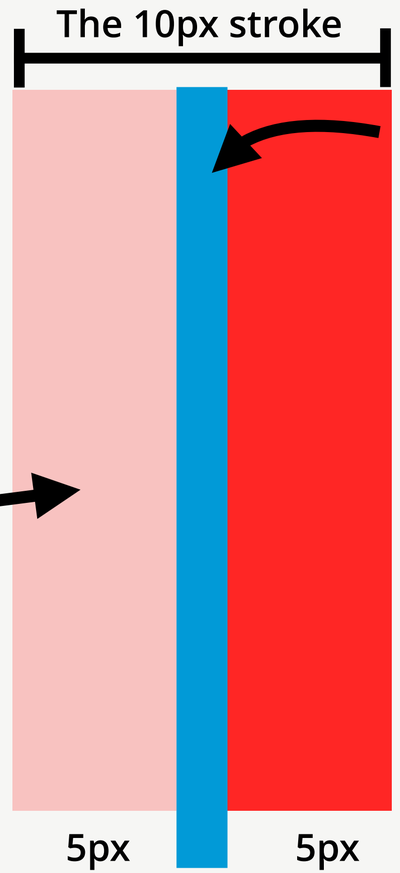
We can create a thicker line with the stroke-width attribute.

```
<rect height="100" width="70" fill="white" stroke="#FF2626" stroke-width="10"/>
```

Investigating the Stroke Cut



Our icons default position starts in the top left corner 0,0.



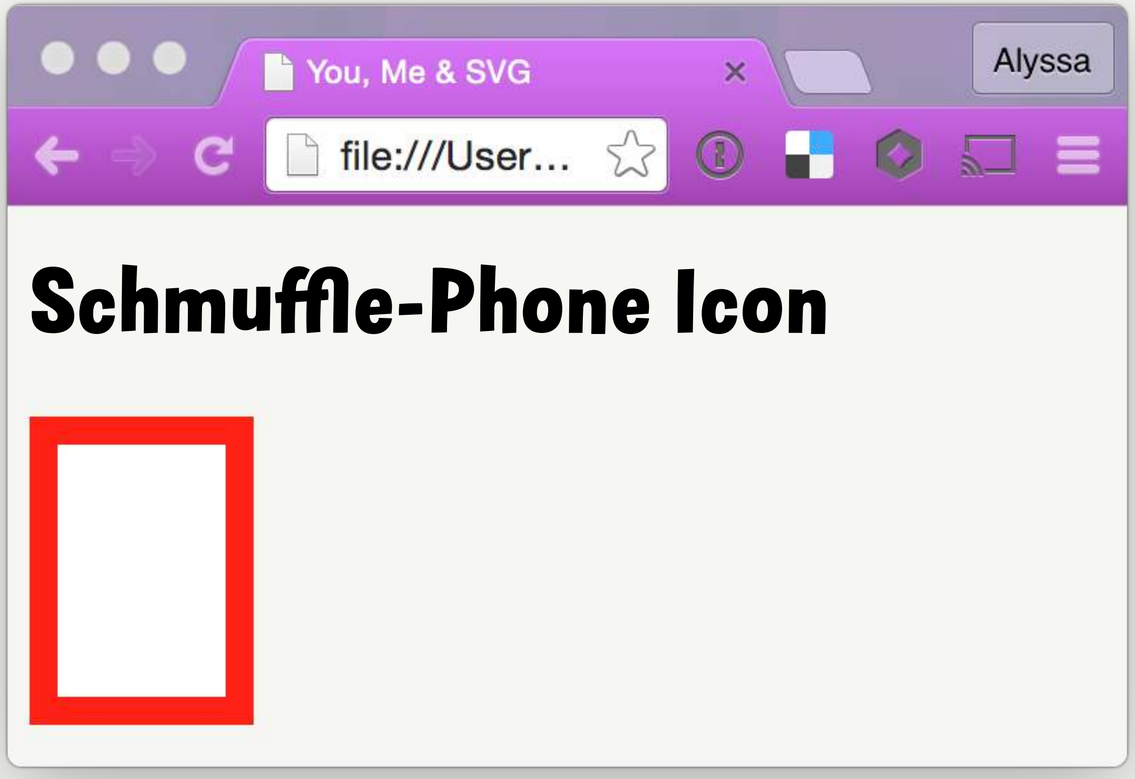
The outline stroke is centered along the rectangle's border.

You, Me & SVG!

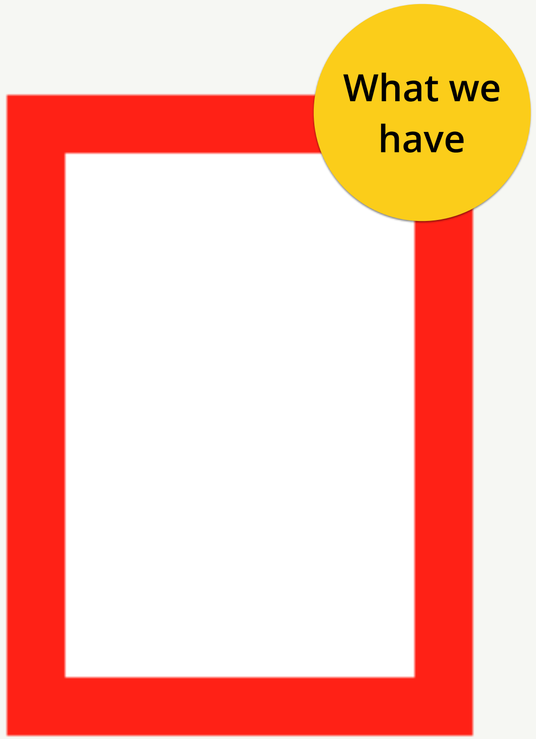
Positioning the Rect

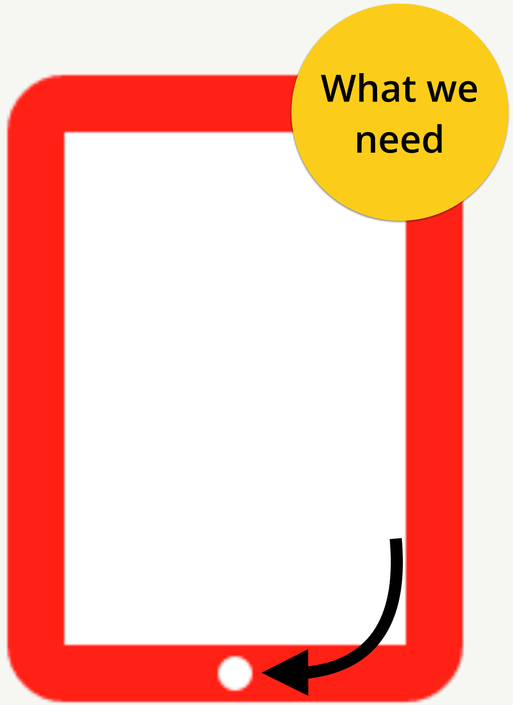
We can move the rectangle origin using x and y.

```
<rect height="100" width="70" fill="white" stroke="#FF2626" stroke-width="10"
      x="5" y="5"/>
```



What’s Left?

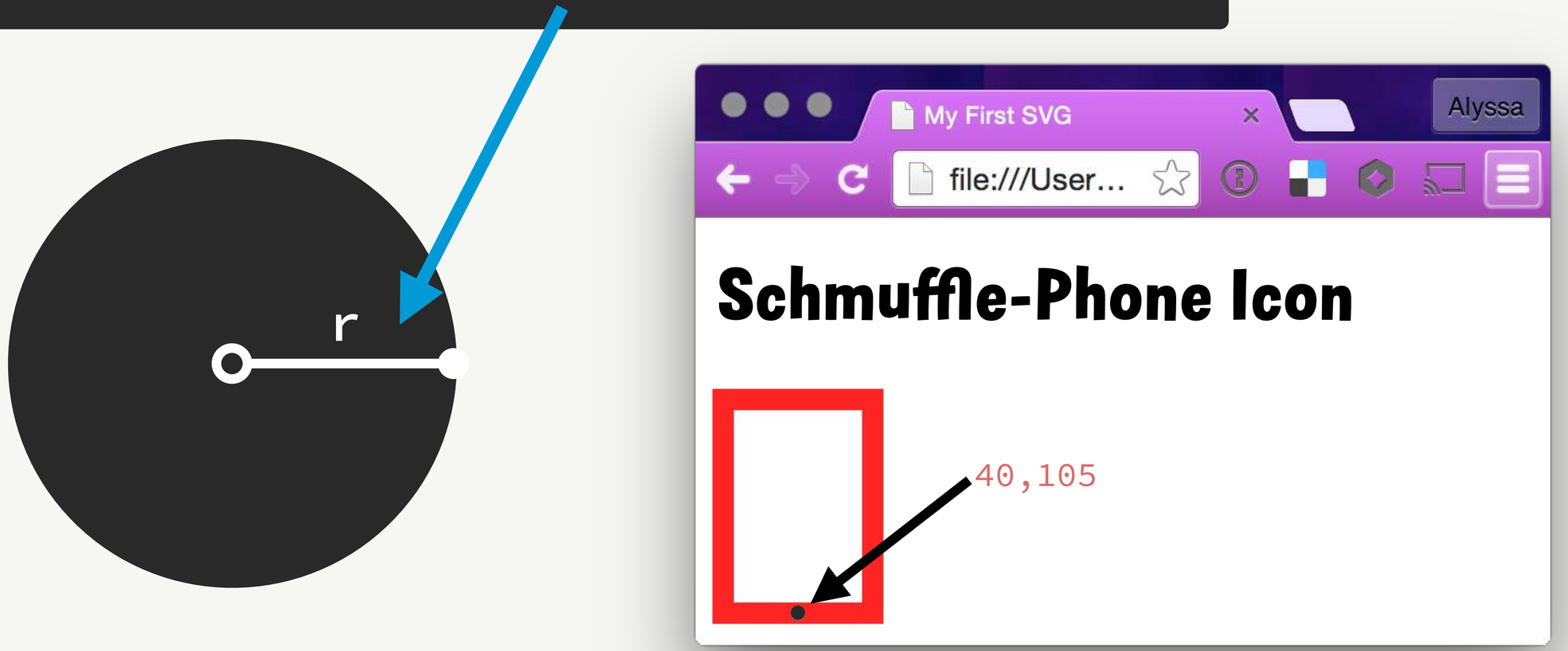




Drawing an SVG Circle

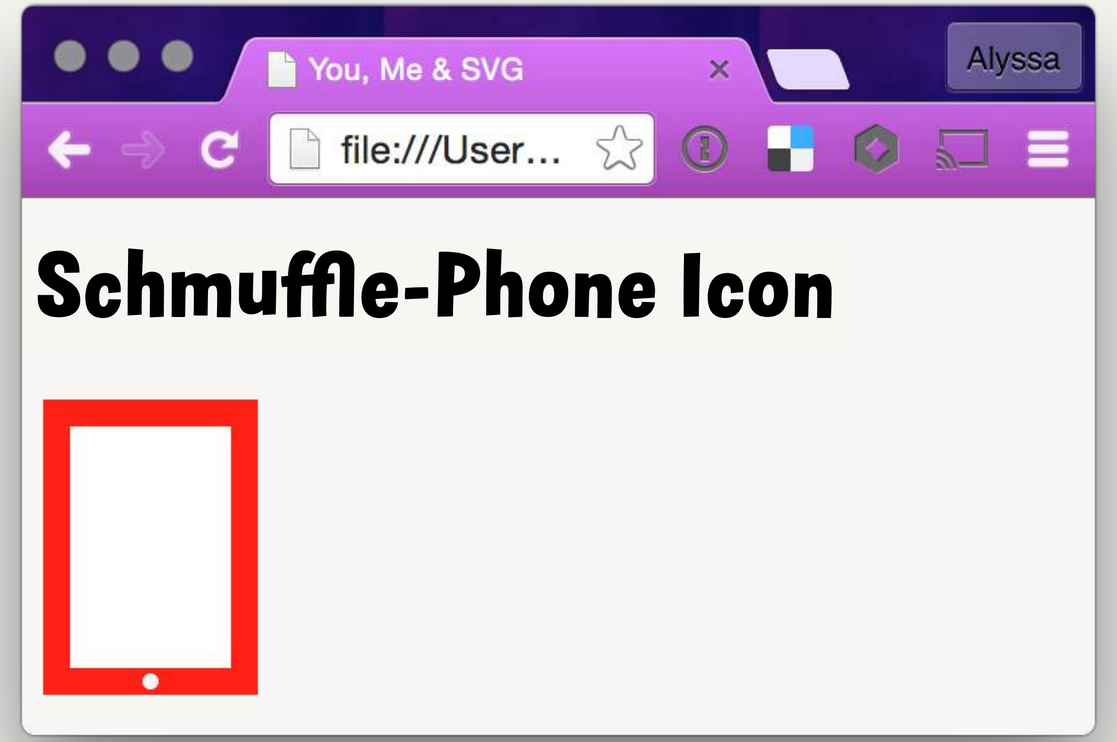
First, we specify the center points cx, cy and specify a radius.

```
<rect height="100" width="70" fill="white" stroke="#FF2626"
    stroke-width="10" x="5" y="5"/>
<circle cx="40" cy="105" r="3"/>
```



Filling the Circle White

```
<circle cx="40" cy="105" r="3" fill="white"/>
```



Now we just need rounded corners on our rect!

But First, We Need to Know About Ellipses

The center points and the fill are the same as the circle element.

```
<ellipse cx="50" cy="50" fill="blue" rx="10" ry="25"/>
```

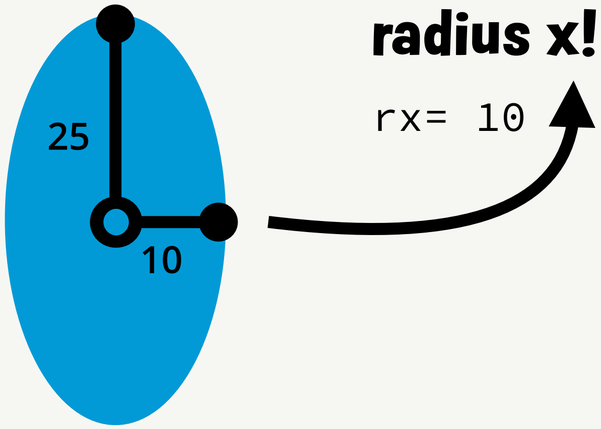
We specify both axes.

radius y!

`ry= 25`

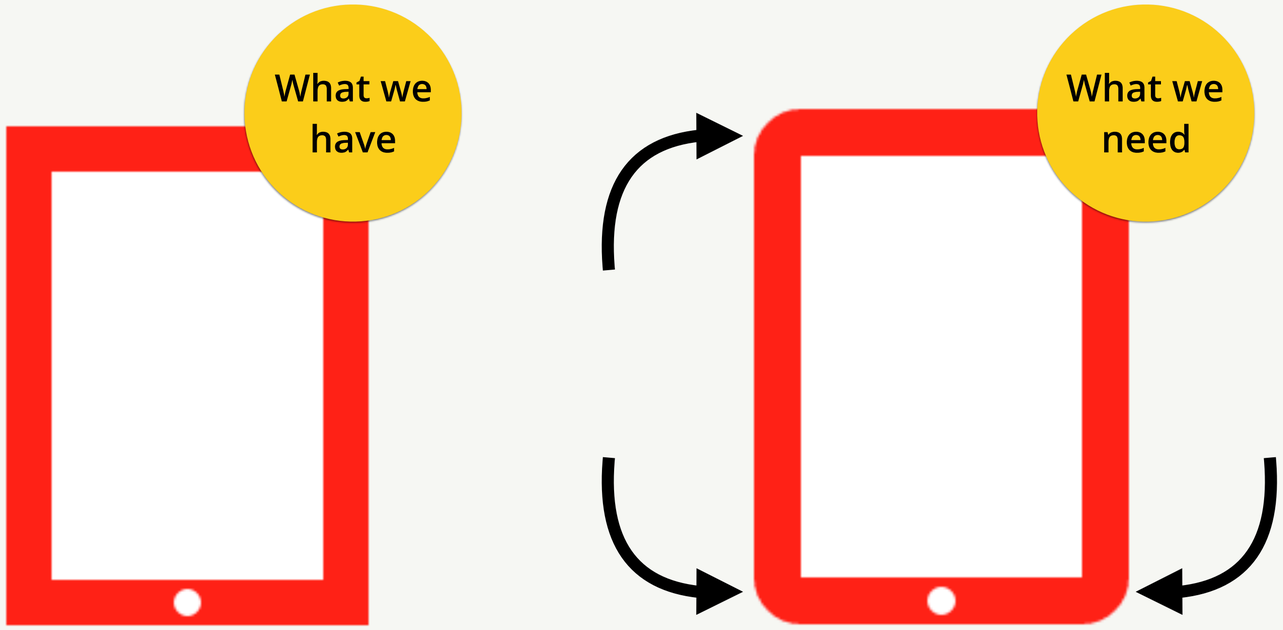
radius x!

`rx= 10`



Unlike a circle, which has the same radius for its x and y axes, ellipse has two different values for x and y axes.

We Still Need to Round the Corners



Rounding Rectangle Corners

To round the corners, we use rx and ry, which set the radiuses on an invisible ellipse.

```
<rect height="100" width="70" fill="white" stroke="#FF2626" stroke-width="10"
x="5" y="5" rx="10" ry="25"/>
```

rx: x-axis radius of the ellipse

ry: y-axis radius of the ellipse

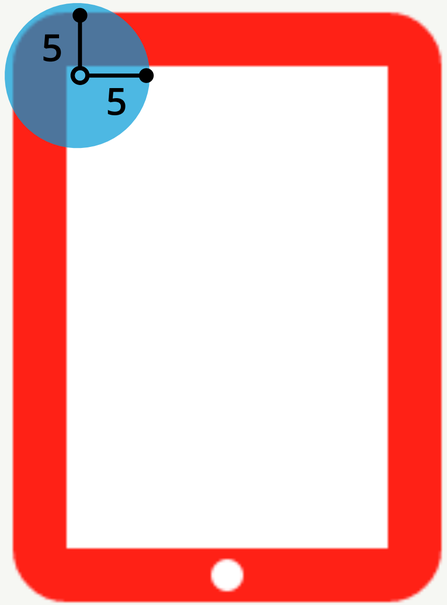
Drawing Our Icon's Rounded Corners

We want our rounded corners to be symmetrical, so we will use the same value for both rx and ry.

```
<rect height="100" width="70" fill="white" stroke="#FF2626"
stroke-width="10" x="5" y="5" rx="5" ry="5"/>
```

same as

```
<rect height="100" width="70" fill="white" stroke="#FF2626"
stroke-width="10" x="5" y="5" rx="5"/>
```



If you only specify rx, the browser will assume the same value for ry!

So Far We Have Used the <img> Tag to Import SVG

index.html

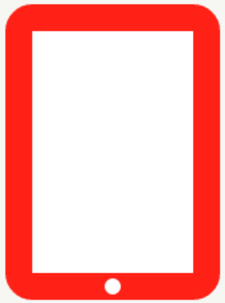
```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <h1>Schmuffle-Screen Icon</h1>
    <img src="phone_icon.svg"/>
  </body>
</html>
```

SVG in an img tag

phone_icon.svg

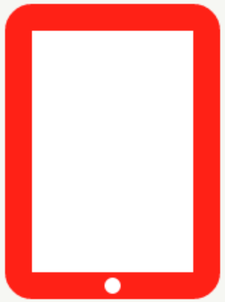
```
<svg height="110" width="80" xmlns="http…" ...>
  <rect height="100" width="70" fill="white" stroke="#FF2626"
        stroke-width="10" x="5" y="5" rx="5"/>
  <circle cx="40" cy="105" r="3" fill="white"/>
</svg>
```

With the image tag, you are able to animate the SVG as a whole...



rotate

scale



animate (transition) on/off screen

Changing SVG's Background?

What if we wanted to scale our phone icon's button? Could we do this with CSS?

Nope! You cannot select inner elements of the SVG when you're including it with an <img>...

But what if we wanted to change the color of our SVG's background?

```
<svg height="110" width="80" xmlns="http…" ... fill="color">
```

Again, nope. We would need to do this through CSS because the SVG tag has no "fill" attribute.

SO HOW CAN WE DO THESE STYLING AND ANIMATING THINGS?!

Another Way to Include Your SVG

There is another, more powerful way to include your SVG! Inline will allow us to control the individual parts of the SVG element.

index.html

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <h1>Schmuffle-Phone Icon</h1>

  </body>
</html>
```

SVG INLINE

```
<svg height="110" width="80" xmlns="http…" ...>
  <rect height="100"
        width="70"
        fill="white"
        stroke="#FF2626"
        stroke-width="10"
        x="5"
        y="5"
        rx="5"/>
  <circle cx="40" cy="105" r="3" fill="white"/>
</svg>
```

SVG Inline HTML

Inline gives us access to the inner elements to iterate on the style or animate with CSS!

index.html

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <h1>Schmuffle-Phone Icon</h1>
      <svg height="110" width="80" xmlns="http…" ...>
        <rect height="100"
              width="70"
              fill="white"
              stroke="#FF2626"
              stroke-width="10"
              x="5"
              y="5"
              rx="5"/>
        <circle cx="40" cy="105" r="3" fill="white"/>
      </svg>
  </body>
</html>
```

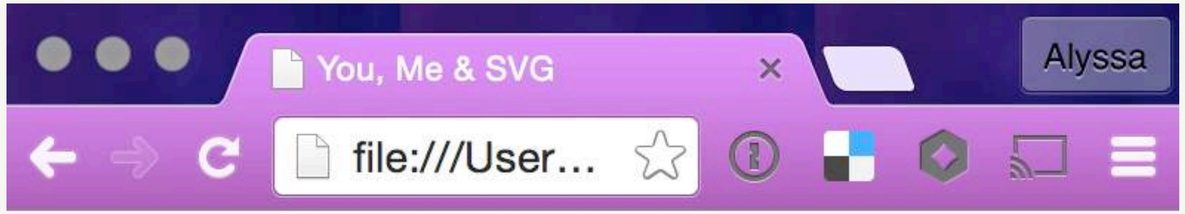
Because the svg is inline ...

SVG Inline With Animation

style.css

```css
circle {
  animation: grow 2s infinite;
  transform-origin: center;
}

@keyframes grow {
  0%    {transform: scale(1);}
  50%   {transform: scale(0.5);}
  100% {transform: scale(1);}
}
```



Schmuffle-Phone Icon

We have access to animate/style individual pieces of the SVG in our CSS!

Challenges